01

Request PUT

Transcripción

[00:00] ¿Qué tal? Bienvenidos a la quinta y última clase de su cursos de Spring Boot API Rest. Hasta el momento ya hemos visto cómo crear registros en nuestra base de datos con el método post y como listarlos con un método get.

[00:18] Ya hemos visto también que si queremos comenzar a ver qué es lo que está sucediendo a nivel de queries en la base de datos, como por ejemplo los inserts, los selects, Hibernate nos puede ayudar imprimiendo pues las queries que son ejecutadas contra la base de datos.

[00:33] Ahora, al inicio de este curso dijimos que íbamos a ver un CRUD, un create, read, update and delete. ¿Qué es lo que nos está faltando? Exacto, los dos últimos, update and delete, y eso es lo que vamos a ver ahora. Vamos a comenzar con el update.

[00:50] Venimos a nuestro requerimiento, venimos a nuestro requerimiento sobre la actualización de médicos. ¿Y qué nos dice? La información permitida para actualizar sería solo nombre, documento y dirección. ¿Qué quiere decir esto? No puedes actualizar, por ejemplo, especialidad, ni email, ni teléfono, por reglas del negocio.

[01:11] Entonces bueno, perfecto. Ya sabemos que nombre, documento o dirección son los únicos campos que podemos actualizar. ¿Y qué más? Bueno, sería básicamente solo eso. Volvemos al código y a diferencia, por ejemplo, de lo que es un Request, como los que hemos visto en Insomnia, un get o un post, vamos a crear un nuevo request para la actualización a nuestro médico.

[01:37] Vamos aquí, igual http://localhost/medicos (http://localhost/medicos). Me faltó el puerto, perdón 8080. Perfecto. Pero a diferencia de los anteriores, este va a ser un tipo put. ¿Por qué? Porque yo voy a enviar un tipo de dato para actualizar otro tipo de dato.

[02:03] Entonces lo que se usa en ese tipo de métodos HTTP es put. Bien, entonces, si yo le doy a enviar, primero voy a iniciar mi servidor. Vamos a iniciar. Vamos a limpiar nuestra consola. Vemos que inició. Voy a enviar este request y me va a decir método no permitido. ¿Por qué? Porque aquí lo que me dice Spring es que el método put no es soportado actualmente y esto está bien.

[02:35] Recordemos que dentro de la misma URL yo puedo tener varios métodos HTTP, por ejemplo, vamos aquí al listado. Recordemos que estos query params son opcionales. Entonces yo puedo hacer simplemente la misma URL si se dan cuenta, pero lo que cambia es el verbo o método HTTP que yo estoy usando para ejecutar ese request, post para registrar, get para obtener y put para actualizar.

[03:04] ¿Qué más necesitamos hasta ahora? Bueno. Vemos que a diferencia de nuestro método para guardar un médico, por ejemplo, entonces venimos aquí en registrarMedico, nosotros le mandamos un datosRegistroMedico, pero si venimos a datosRegistroMedico, vamos a ver que no podemos mandar nombre en blanco, ni email, ni teléfono en blanco y básicamente ningún parámetro en blanco.

[03:30] Entonces el DTO que vamos a usar no puede ser este mismo. Tenemos que crear otro DTO. Entonces hay dos cosas que podemos hacer ahora. Vamos a public y de igual forma un void y vamos a decirle actualizarMedico, sin parámetros por ahora y este va a tener un método put. Y como ya saben, aquí va putMapping.

[04:03] Y listo, porque estamos mapeando la misma URL. Segundo, ¿qué es lo que necesitamos? Ya vimos que la actualización debería ser más o menos como

el registro. Pero en el registro yo recibo los datos de registro médico y para actualizar, yo debo saber qué médico es el que debo actualizar.

[04:23] Si vamos a listar los médicos, por ejemplo, estamos aquí, ahora sí, mi servidor demoró un poco en venir. Vemos que, por ejemplo yo tengo aquí dos Diego López. Uno tiene un documento diferente del otro, podemos decir que son homónimos, tienen el mismo nombre, pero su número de documento, pues es diferente.

[04:49] Ahora hay un correo también. ¿Yo cómo sé cuál de los Diego López, yo quiero actualizar? Lo que se hace normalmente en estos casos es usar el id, si vamos a la base de datos, yo debería aquí mi base de datos, voy a abrir mi tabla médicos, ustedes van a ver que aquí hay un elemento único para cada registro que es el id.

[05:12] A nivel de API, lo que hacemos normalmente es retornar ese id al cliente, de modo que el cliente por ejemplo si tú muestras una lista, vamos a la tabla anterior. Si tú en la lista, por ejemplo, si en la aplicación móvil yo le doy clic a Adriano Moreira, por ejemplo internamente, ya deberías saber cuál es el id de Adriano Moreira.

[05:38] Y enviarme ese id como parámetro para yo decir: “correcto, yo necesito modificar el elemento con este id”, que es un elemento único a nivel de base de datos. Y eso es lo que vamos a hacer en este momento, voy a incluir el id en el payload para listar los médicos. ¿Cómo hago eso? Muy simple.

[06:01] Aquí vamos a ir primero que todo a datosListadoMedico que es nuestro DTO. Voy a cerrar aquí. Y lo que voy a decirle aquí es que voy a recibir un Long id, y aquí yo le voy a decir medico.getId(). Y listo, eso sería lo que yo necesito para mostrar el id en mi listado de médicos. Voy a guardar, voy a esperar que se refresque mi servidor. Vemos que ya refrescó, y vamos a ver si funciona mi método.

[06:40] Le voy a dar otra vez enviar y ahora ya tengo el id aquí. Por lo tanto, yo ya sé que Diego López, que yo voy a modificar aquí es el Diego López con el id 5. Perfecto. Entonces, ¿qué más necesitamos ahora? Los parámetros. ¿Por qué? Porque yo necesito de alguna forma enviarle los datos que yo quiero modificar. ¿Cómo hago eso? Lo vamos a ver en el siguiente video.